Raymond Cano
ELEC 677
HW 0

Task 1:
packages in environment at /Users/raymond/anaconda2:
#
_nb_ext_conf 0.2.0 py27_0
alabaster 0.7.8 py27_0
anaconda 4.1.1 np111py27_0
anaconda-client 1.4.0 py27_0
anaconda-navigator 1.2.1 py27_0
appnope 0.1.0 py27_0
appscript 1.0.1 py27_0
argcomplete 1.0.0 py27_1
astropy 1.2.1 np111py27_0
babel 2.3.3 py27_0
backports 1.0 py27_0
backports_abc 0.4 py27_0
beautifulsoup4 4.4.1 py27_0
bitarray 0.8.1 py27_0
blaze 0.10.1 py27_0
bokeh 0.12.0 py27_0
boto 2.40.0 py27_0
bottleneck 1.1.0 np111py27_0
cdecimal 2.3 py27_2
cffi 1.6.0 py27_0
chest 0.2.3 py27_0
click 6.6 py27_0
cloudpickle 0.2.1 py27_0
clyent 1.2.2 py27_0
colorama 0.3.7 py27_0
conda 4.1.6 py27_0
conda-build 1.21.3 py27_0
conda-env 2.5.1 py27_0
configobj 5.0.6 py27_0
configparser 3.5.0b2 py27_1
contextlib2 0.5.3 py27_0
cryptography 1.4 py27_0
curl 7.49.0 0
cycler 0.10.0 py27_0
cython 0.24 py27_0
cytoolz 0.8.0 py27_0
dask 0.10.0 py27_0
datashape 0.5.2 py27_0
decorator 4.0.10 py27_0
dill 0.2.5 py27_0
docutils 0.12 py27_2
dynd-python 0.7.2 py27_0

entrypoints	0.2.2	py27_0
enum34	1.1.6	py27_0
et_xmlfile	1.0.1	py27_0
fastcache	1.0.2	py27_1
flask	0.11.1	py27_0
flask-cors	2.1.2	py27_0
freetype	2.5.5	1
funcsigs	1.0.2	py27_0
functools32	3.2.3.2	py27_0
futures	3.0.5	py27_0
get_terminal_size	1.0.0	py27_0
gevent	1.1.1	py27_0
greenlet	0.4.10	py27_0
grin	1.2.1	py27_3
h5py	2.6.0	np111py27_1
hdf5	1.8.16	0
heapdict	1.0.0	py27_1
idna	2.1	py27_0
imagesize	0.7.1	py27_0
ipaddress	1.0.16	py27_0
ipykernel	4.3.1	py27_0
ipython	4.2.0	py27_1
ipython_genutils	0.1.0	py27_0
ipywidgets	4.1.1	py27_0
itsdangerous	0.24	py27_0
jbig	2.1	0
jdcal	1.2	py27_1
jedi	0.9.0	py27_1
jinja2	2.8	py27_1
jpeg	8d	1
jsonschema	2.5.1	py27_0
jupyter	1.0.0	py27_3
jupyter_client	4.3.0	py27_0
jupyter_console	4.1.1	py27_0
jupyter_core	4.1.0	py27_0
libdynd	0.7.2	0
libpng	1.6.22	0
libtiff	4.0.6	2
libxml2	2.9.2	0
libxslt	1.1.28	2
llvmlite	0.11.0	py27_0
locket	0.2.0	py27_1
lxml	3.6.0	py27_0
markupsafe	0.23	py27_2
matplotlib	1.5.1	np111py27_0
mistune	0.7.2	py27_1
mkl	11.3.3	0
mkl-service	1.1.2	py27_2
mpmath	0.19	py27_1

multipledispatch 0.4.8 py27_0
nb_anacondacloud 1.1.0 py27_0
nb_conda 1.1.0 py27_0
nb_conda_kernels 1.0.3 py27_0
nbconvert 4.2.0 py27_0
nbformat 4.0.1 py27_0
nbpresent 3.0.2 py27_0
networkx 1.11 py27_0
nltk 3.2.1 py27_0
nose 1.3.7 py27_1
notebook 4.2.1 py27_0
numba 0.26.0 np111py27_0
numexpr 2.6.0 np111py27_0
numpy 1.11.1 py27_0
odo 0.5.0 py27_1
openpyxl 2.3.2 py27_0
openssl 1.0.2h 1
pandas 0.18.1 np111py27_0
partd 0.3.4 py27_0
path.py 8.2.1 py27_0
pathlib2 2.1.0 py27_0
patsy 0.4.1 py27_0
pep8 1.7.0 py27_0
pexpect 4.0.1 py27_0
pickleshare 0.7.2 py27_0
pillow 3.2.0 py27_1
pip 8.1.2 py27_0
ply 3.8 py27_0
psutil 4.3.0 py27_0
ptyprocess 0.5.1 py27_0
py 1.4.31 py27_0
pyasn1 0.1.9 py27_0
pyaudio 0.2.7 py27_0
pycosat 0.6.1 py27_1
pycparser 2.14 py27_1
pycrypto 2.6.1 py27_4
pycurl 7.43.0 py27_0
pyflakes 1.2.3 py27_0
pygments 2.1.3 py27_0
pyopenssl 0.16.0 py27_0
pyparsing 2.1.4 py27_0
pyqt 4.11.4 py27_3
pytables 3.2.2 np111py27_4
pytest 2.9.2 py27_0
python 2.7.12 1
python-dateutil 2.5.3 py27_0
python.app 1.2 py27_4
pytz 2016.4 py27_0
pyyaml 3.11 py27_4

```
pyzmq                 15.2.0               py27_1
qt                    4.8.7                3
qtconsole             4.2.1                py27_0
qtpy                  1.0.2                py27_0
readline              6.2                  2
redis                 3.2.0                0
redis-py              2.10.5               py27_0
requests              2.10.0               py27_0
rope                  0.9.4                py27_1
ruamel_yaml           0.11.7               py27_0
scikit-image          0.12.3               np111py27_1
scikit-learn          0.17.1               np111py27_2
scipy                 0.17.1               np111py27_1
setuptools            23.0.0               py27_0
simplegeneric         0.8.1                py27_1
singledispatch        3.4.0.3              py27_0
sip                   4.16.9               py27_0
six                   1.10.0               py27_0
snowballstemmer       1.2.1                py27_0
sockjs-tornado        1.0.3                py27_0
sphinx                1.4.1                py27_0
sphinx_rtd_theme      0.1.9                py27_0
spyder                2.3.9                py27_0
sqlalchemy            1.0.13               py27_0
sqlite                3.13.0               0
ssl_match_hostname    3.4.0.2              py27_1
statsmodels           0.6.1                np111py27_1
sympy                 1.0                  py27_0
terminado             0.6                  py27_0
tk                    8.5.18               0
toolz                 0.8.0                py27_0
tornado               4.3                  py27_1
traitlets             4.2.1                py27_0
unicodecsv            0.14.1               py27_0
werkzeug              0.11.10              py27_0
wheel                 0.29.0               py27_0
xlrd                  1.0.0                py27_0
xlsxwriter            0.9.2                py27_0
xlwings               0.7.2                py27_0
xlwt                  1.1.2                py27_0
xz                    5.2.2                0
yaml                  0.1.6                0
zlib                  1.2.8                3
```

Task 2:
Note: This was partially done because the activity was miticulous and very lengthy.

```
# Script
a = np.zeros((1, 15))
```

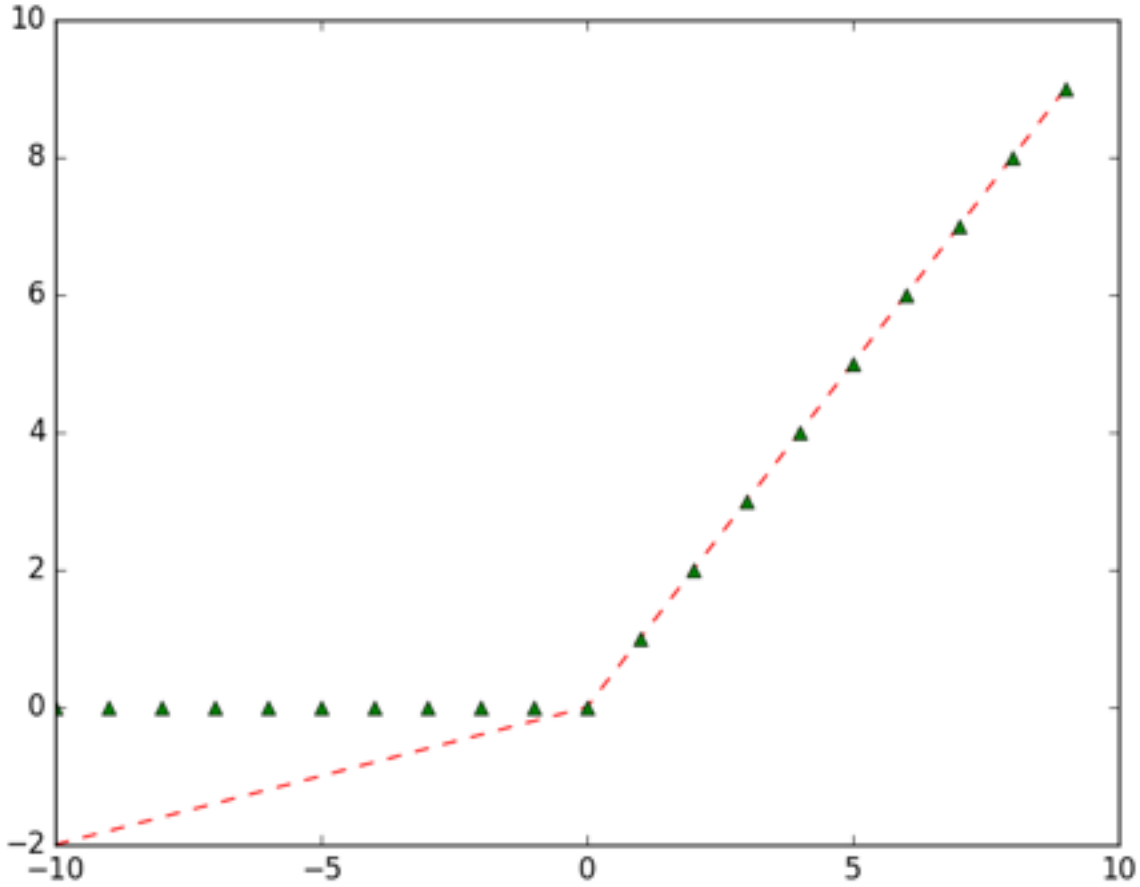
```
a = np.arange(15).reshape(3,5)

print a
print a.ndim
print a.size
print a.shape
print a.shape[0]
print np.array([[1., 2., 3.], [4.,5.,6.,]])
v = np.array([1,2])
q = np.array([2,3])
n = np.array([3,4])
m = np.array([4,5])
stacked = np.vstack([np.hstack([v, q]), np.hstack([n, m])])
print stacked
print a[-1]
print a[2, 3]
print stacked[1,:]
print a[:1]
print a[-2:]
print a[0:1][:, 2:4]
print a[np.ix_([1,2],[0,2])]
print a[:, 1:4:2]
print a[::2, :]
print a[::-1, :]
print a[:, ::-1]
print a[np.r_[:len(a), 0]]
print a.T
print a.conj().T
b = np.arange(15).reshape(5,3)
print a.dot(b)
print a * b.T
print a / (b.T + 1.0)
print a ** 2
print (a>4)
print np.nonzero(a> 11)
print a[:, np.nonzero(a[:,2] > 3)[0]]
print np.arange(1., 11.)
print np.arange(0,9)
print a.flatten()
print np.zeros((3,4))
print np.zeros((3,4,5))
print np.ones((3,4))
print np.eye(3)
print np.diag(a)
print np.diag(a, 0)
print np.random.rand(5,4)
print np.linspace(1,9,5)
print np.mgrid[0:9., 0:6]
```

```
# Output
[[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]]
2
15
(3, 5)
3
[[ 1.  2.  3.]
 [ 4.  5.  6.]]
[[1 2 2 3]
 [3 4 4 5]]
[10 11 12 13 14]
13
[3 4 4 5]
[[0 1 2 3 4]]
[[ 5  6  7  8  9]
 [10 11 12 13 14]]
[[2 3]]
[[ 5  7]
 [10 12]]
[[ 1  3]
 [ 6  8]
 [11 13]]
[[ 0  1  2  3  4]
 [10 11 12 13 14]]
[[10 11 12 13 14]
 [ 5  6  7  8  9]
 [ 0  1  2  3  4]]
[[ 4  3  2  1  0]
 [ 9  8  7  6  5]
 [14 13 12 11 10]]
[[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]
 [ 0  1  2  3  4]]
[[ 0  5 10]
 [ 1  6 11]
 [ 2  7 12]
 [ 3  8 13]
 [ 4  9 14]]
[[ 0  5 10]
 [ 1  6 11]
 [ 2  7 12]
 [ 3  8 13]
 [ 4  9 14]]
[[ 90 100 110]
 [240 275 310]
 [390 450 510]]
```

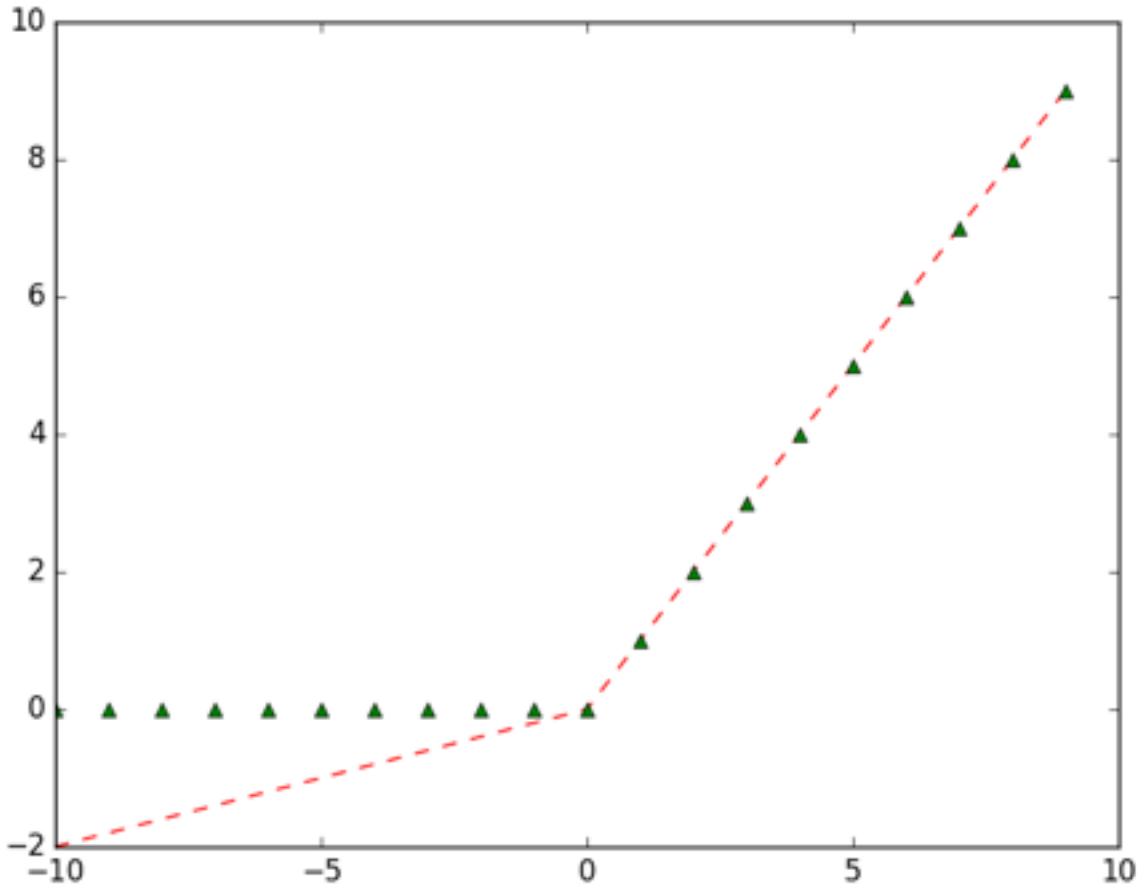
```
[[  0   3  12  27  48]
 [  5  24  49  80 117]
 [ 20  55  96 143 196]]
[[ 0.          0.25        0.28571429  0.3         0.30769231]
 [ 2.5         1.2         0.875       0.72727273  0.64285714]
 [ 3.33333333  1.83333333  1.33333333  1.08333333  0.93333333]]
[[  0   1   4   9  16]
 [ 25  36  49  64  81]
 [100 121 144 169 196]]
[[False False False False False]
 [ True  True  True  True  True]
 [ True  True  True  True  True]]
(array([2, 2, 2]), array([2, 3, 4]))
[[ 1  2]
 [ 6  7]
 [11 12]]
[  1.   2.   3.   4.   5.   6.   7.   8.   9.  10.]
[0 1 2 3 4 5 6 7 8]
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14]
[[ 0.  0.  0.  0.]
 [ 0.  0.  0.  0.]
 [ 0.  0.  0.  0.]]
[[[ 0.  0.  0.  0.  0.]
  [ 0.  0.  0.  0.  0.]
  [ 0.  0.  0.  0.  0.]
  [ 0.  0.  0.  0.  0.]]

 [[ 0.  0.  0.  0.  0.]
  [ 0.  0.  0.  0.  0.]
  [ 0.  0.  0.  0.  0.]
  [ 0.  0.  0.  0.  0.]]

 [[ 0.  0.  0.  0.  0.]
  [ 0.  0.  0.  0.  0.]
  [ 0.  0.  0.  0.  0.]
  [ 0.  0.  0.  0.  0.]]]
[[ 1.  1.  1.  1.]
 [ 1.  1.  1.  1.]
 [ 1.  1.  1.  1.]]
[[ 1.  0.  0.]
 [ 0.  1.  0.]
 [ 0.  0.  1.]]
[ 0  6 12]
[ 0  6 12]
[[ 0.71495127  0.26944766  0.63765697  0.21562208]
 [ 0.27989592  0.04276152  0.24996509  0.82447833]
 [ 0.28670426  0.93043583  0.89702219  0.92850914]
 [ 0.40774586  0.63096233  0.70683138  0.56094226]
 [ 0.38645777  0.66895346  0.28924643  0.64798422]]
```

```
[ 1.  3.  5.  7.  9.]
[[[ 0.  0.  0.  0.  0.  0.]
  [ 1.  1.  1.  1.  1.  1.]
  [ 2.  2.  2.  2.  2.  2.]
  [ 3.  3.  3.  3.  3.  3.]
  [ 4.  4.  4.  4.  4.  4.]
  [ 5.  5.  5.  5.  5.  5.]
  [ 6.  6.  6.  6.  6.  6.]
  [ 7.  7.  7.  7.  7.  7.]
  [ 8.  8.  8.  8.  8.  8.]]

 [[ 0.  1.  2.  3.  4.  5.]
  [ 0.  1.  2.  3.  4.  5.]
  [ 0.  1.  2.  3.  4.  5.]
  [ 0.  1.  2.  3.  4.  5.]
  [ 0.  1.  2.  3.  4.  5.]
  [ 0.  1.  2.  3.  4.  5.]
  [ 0.  1.  2.  3.  4.  5.]
  [ 0.  1.  2.  3.  4.  5.]
  [ 0.  1.  2.  3.  4.  5.]]]
```

Task 3:



Task 4:



Relu vs. Leaky Relu

Task 5:
https://bitbucket.org/cheongcano/

Task 6:
Void Return Value

Task 7:
Bad Coding Habit: I tend to turn routines that are only called once into subroutines. I've been told this is an unnecessary layer of abstraction and I should only look to modularize code that can needs to be used more than once. To improve on this, I should better plan out what needs to be used more than once and thus, what needs to be turned into a subroutine.